VXI

VXI-8340 Series
User Manual

May 2001 Edition
Part Number 370358B-01

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 794 0100

Worldwide Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011,
Canada (Calgary) 403 274 9391, Canada (Montreal) 514 288 5722, Canada (Ottawa) 613 233 5949,
Canada (Québec) 514 694 8521, Canada (Toronto) 905 785 0085, China (Shanghai) 021 6555 7838,
China (ShenZhen) 0755 3904939, Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24,
Germany 089 741 31 30, Greece 30 1 42 96 427, Hong Kong 2645 3186, India 91805275406,
Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Malaysia 603 9596711,
Mexico 5 280 7625, Netherlands 0348 433466, New Zealand 09 914 0488, Norway 32 27 73 00,
Poland 0 22 528 94 06, Portugal 351 1 726 9011, Singapore 2265886, Spain 91 640 0085,
Sweden 08 587 895 00, Switzerland 056 200 51 51, Taiwan 02 2528 7227, United Kingdom 01635 523545

For further support information, see the *Technical Support Resources* appendix. To comment on the documentation, send e-mail to techpubs@ni.com.

Copyright © 2001 National Instruments Corporation. All rights reserved.

Important Information

Warranty

The National Instruments VXI-8340 is warranted against defects in materials and workmanship for a period of one year from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

ComponentWorks™, CVI™, LabVIEW™, National Instruments™, ni.com™, NI-DAQ™, NI-VISA™, and NI-VXI™ are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Compliance

FCC/Canada Radio Frequency Interference Compliance*

Determining FCC Class

The Federal Communications Commission (FCC) has rules to protect wireless communications from interference. The FCC places digital electronics into two classes. These classes are known as Class A (for use in industrial-commercial locations only) or Class B (for use in residential or commercial locations). Depending on where it is operated, this product could be subject to restrictions in the FCC rules. (In Canada, the Department of Communications (DOC), of Industry Canada, regulates wireless interference in much the same way.)

Digital electronics emit weak signals during normal operation that can affect radio, television, or other wireless products. By examining the product you purchased, you can determine the FCC Class and therefore which of the two FCC/DOC Warnings apply in the following sections. (Some products may not be labeled at all for FCC; if so, the reader should then assume these are Class A devices.)

FCC Class A products only display a simple warning statement of one paragraph in length regarding interference and undesired operation. Most of our products are FCC Class A. The FCC rules have restrictions regarding the locations where FCC Class A products can be operated.

FCC Class B products display either a FCC ID code, starting with the letters **EXN**, or the FCC Class B compliance mark that appears as shown here on the right.

Consult the FCC web site `http://www.fcc.gov` for more information.

FCC/DOC Warnings

This equipment generates and uses radio frequency energy and, if not installed and used in strict accordance with the instructions in this manual and the CE Mark Declaration of Conformity**, may cause interference to radio and television reception. Classification requirements are the same for the Federal Communications Commission (FCC) and the Canadian Department of Communications (DOC).

Changes or modifications not expressly approved by National Instruments could void the user's authority to operate the equipment under the FCC Rules.

Class A

Federal Communications Commission

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Canadian Department of Communications

This Class A digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

Cet appareil numérique de la classe A respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

Class B

Federal Communications Commission

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Canadian Department of Communications

This Class B digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

Cet appareil numérique de la classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

Compliance to EU Directives

Readers in the European Union (EU) must refer to the Manufacturer's Declaration of Conformity (DoC) for information** pertaining to the CE Mark compliance scheme. The Manufacturer includes a DoC for most every hardware product except for those bought for OEMs, if also available from an original manufacturer that also markets in the EU, or where compliance is not required as for electrically benign apparatus or cables.

To obtain the DoC for this product, click **Declaration of Conformity** at `ni.com/hardref.nsf/`. This web site lists the DoCs by product family. Select the appropriate product family, followed by your product, and a link to the DoC appears in Adobe Acrobat format. Click the Acrobat icon to download or read the DoC.

* Certain exemptions may apply in the USA, see FCC Rules §15.103 **Exempted devices**, and §15.105(c). Also available in sections of CFR 47.

** The CE Mark Declaration of Conformity will contain important supplementary information and instructions for the user or installer.

Contents

About This Manual

How To Use the Manual Set ix
Conventions x
Related Documentation xi

Chapter 1
Introduction

What You Need to Get Started 1-1
VXI-8340 Series Interface Kit Overview 1-2
Hardware Description 1-2
    VXI-8340 Series Front Panel Features 1-3
    Advanced Configuration Options 1-3
Software Description 1-4
    Software Configurations 1-4
    National Instruments Application Software 1-5

Chapter 2
Installation and Configuration

Installing the Software 2-1
    Install the NI-VISA and NI-VXI Software 2-1
    Complete the Software Installation 2-2
Installing the Hardware 2-3
    Install Your PCI-8330 Series Interface Board 2-3
    Install Your VXI-8340 Series Interface Module 2-4
    Connect the Cable 2-5
    Power on the System 2-5
Software Configuration and Verification 2-6
    Run MAX and Resman 2-6
Configuring the Hardware (Optional) 2-6
Using the Request Offline/Go Online Feature 2-7

Chapter 3
Developing Your Application

Configuration 3-1
Device Interaction 3-2

Programming with VXI 3-3
Compiler Symbols for NI-VXI 3-5
Window Mapping 3-6
Address Space 3-7
Number of MITE Windows 3-7
Memory for Allocating Data Structures 3-7
MITE DMA 3-7
Shared Memory 3-8
Enhancements to the NI-VXI Software 3-9
Compatibility 3-9
System Configuration Functions 3-9
Low-Level VXIbus Access Functions 3-9
High-Level VXIbus Access Functions 3-9
Local Resource Access Functions 3-10
Debugging 3-10

Appendix A
Specifications

Appendix B
Default Settings

Appendix C
Advanced Hardware Configuration Settings

Appendix D
Power Sequencing Requirements

Appendix E
Common Questions

Appendix F
Technical Support Resources

Glossary

Index

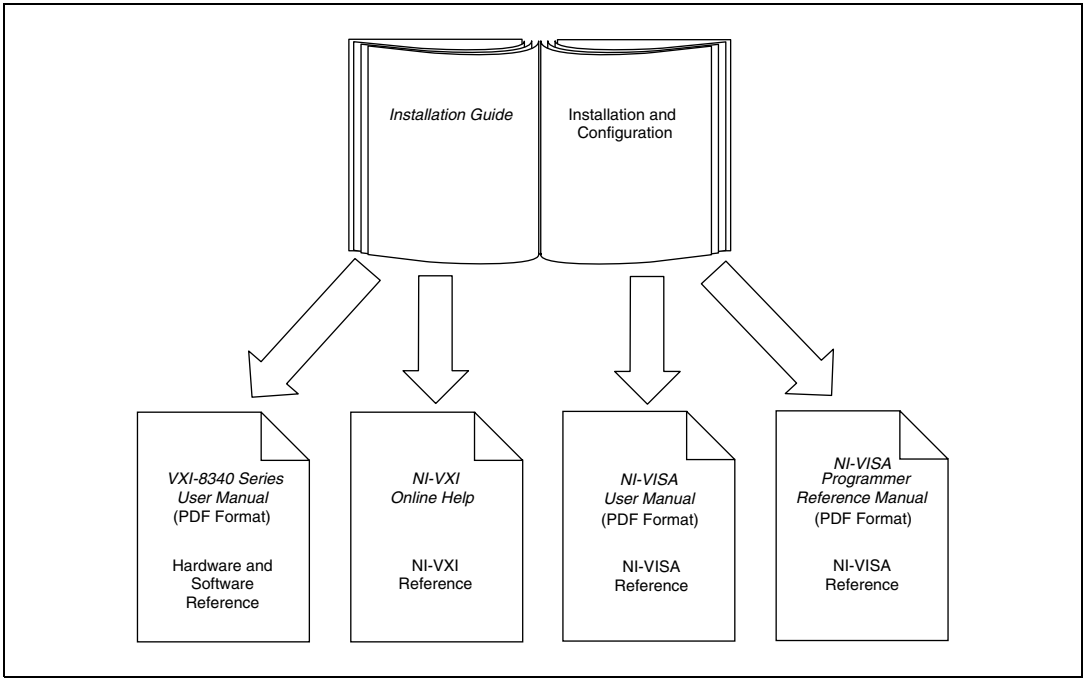
About This Manual

This manual contains instructions for installing and configuring any of six National Instruments VXI-8340 series kits:

- VXI-8340 single-port module/PCI-8330 board interface kit
- VXI-8345 single-port module/PCI-8335 board interface kit
- VXI-8340 single-port module-only kit
- VXI-8340 dual-port module-only kit
- VXI-8345 single-port module-only kit
- VXI-8345 dual-port module-only kit

The VXI-8340 series is a VXI*plug&play*-compliant interface that gives external PCI-based computers the capabilities of embedded VXI controllers.

How To Use the Manual Set



This user manual contains an overview of the VXI-8340 series and PCI-8330 series hardware and the NI-VISA and NI-VXI software, guides you through setting up your kit, and helps you get started with application development. You can also use this manual as a reference for the hardware and software default settings and to find answers to commonly asked questions.

When you have successfully set up your system, you can begin to develop applications in NI-VISA or NI-VXI. NI-VISA online manuals are included in your kit in the form of Adobe Acrobat portable document format (PDF) files. The *NI-VXI Online Help* presents the concepts of VXI and detailed explanations of the NI-VXI functions. Study the descriptions of each function given in the online help to fully understand the purpose and syntax of each function.

The Acrobat manuals and their installed locations are as follows:

- The *NI-VISA User Manual* describes how to program using NI-VISA: **Start»Programs»VXIpnp»NI-VISA User Manual**
- The *NI-VISA Programmer Reference Manual* describes in detail the attributes, events, and operations you use in NI-VISA: **Start» Programs»VXIpnp»NI-VISA Programmer Reference Manual**

Use the Acrobat Reader program to open the PDF files.

Note You can download Acrobat Reader from the Adobe Web site at `www.adobe.com`.

Conventions

The following conventions appear in this manual:

» The **»** symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

This icon denotes a note, which alerts you to important information.

This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.

bold Bold text denotes items that you must select or click on in the software, such as menu items and dialog box options. Bold text also denotes parameter names.

italic	Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text that is a placeholder for a word or value that you must supply.
`monospace`	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.
`monospace bold`	Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.
`monospace italic`	Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Related Documentation

The following documents contain information that you may find helpful as you read this manual:

- ANSI/IEEE Standard 1014-1987, *IEEE Standard for a Versatile Backplane Bus: VMEbus*
- ANSI/IEEE Standard 1155-1998, *IEEE VMEbus Extensions for Instrumentation: VXIbus*
- ANSI/VITA 1-1994, *VME64*
- *PCI Local Bus Specification*, Revision 2.1, PCI Special Interest Group
- VXI-6, *VXIbus Mainframe Extender Specification*, Rev. 2.0, VXIbus Consortium

1

Introduction

This chapter describes your VXI-8340 series interface kit, lists what you need to get started, and includes a brief description of the hardware and software. This manual uses the term *VXI-8340 series* when information applies to both the VXI-8340 and VXI-8345.

The VXI-8340 series interface kit links a PCI-based computer to the VXIbus using the high-speed Multisystem eXtension Interface (MXI-3) bus. This interface makes your computer function as if it were plugged directly into the VXI backplane, giving your external computer the capability of an embedded computer.

Your kit contains a National Instruments VXI-8340 series interface module, which plugs into your VXI mainframe. It may also include a MXI-3 cable and a PCI-8330 series interface module, which plugs into your PCI-based computer. This manual uses the term *PCI-8330 series* when information applies to both the PCI-8330 and PCI-8335.

Your kit includes the NI-VISA and NI-VXI bus interface software, which is fully VXI*plug&play* compliant. NI-VISA and NI-VXI are the National Instruments implementation of the VISA I/O software standard on which all VXI*plug&play* software components are based. This manual uses the term *NI-VISA/NI-VXI* when software information applies to both NI-VISA and NI-VXI, and the term *Windows* when information applies to Windows 2000/NT/Me/98. (The VXI-8340 series is not compatible with Windows 95.)

What You Need to Get Started

- ❑ A PCI-based computer running Windows
- ❑ VXIbus mainframe
- ❑ VXI-8340 series interface module
- ❑ PCI-8330 series interface module (not included in VXI-8340 series module-only kits)

- ❑ Copper or fiber-optic MXI-3 cable (not included in VXI-8340 series module-only kits)
- ❑ National Instruments *NI-VXI v. 3.1 for Windows* CD

VXI-8340 Series Interface Kit Overview

The interface kits described in this manual link a PCI bus-based computer to the VXIbus using MXI-3.

All VXI-8340 series kits include the NI-VISA/NI-VXI software for Windows and a VXI-8340 series module. The kits may also contain a PCI-8330 series module and a MXI-3 cable.

A PCI bus computer connected to a VXI-8340 series interface can function as a VXI Commander and Resource Manager. The VXI-8340 series interface kit allows your computer to function as if it were plugged directly into the VXI backplane as an embedded CPU module.

The software included with the kits is for *x*86-based computers only.

Hardware Description

The VXI-8340 series module is a C-size VXIbus device with VXIbus Slot 0 capability so that it can reside in any slot of a C-size or D-size chassis. If you enable this capability, the VXI-8340 series can automatically determine whether it is in VXI Slot 0.

Note D-size VXI mainframes have connections for a P3 connector. The VXI-8340 series, however, does not have this connector and, if configured as a Slot 0 controller, cannot provide the necessary control for VXI devices that need P3 support.

Caution An improper Slot 0 setting may damage the VXI-8340 series module and/or the VXI chassis.

The VXI-8340 series links the computer to the VXIbus and converts PCI data transfers into VXIbus data transfers and vice versa.

The PCI-8330 is a PCI bus extender, bridging your PCI bus across a serial link.

VXI-8340 Series Front Panel Features

The VXI-8340 series has the following front panel features:

- MXI IN port for connecting the VXI-8340 series to a PCI bus-equipped PC or the MXI OUT port of another VXI-8340 series
- MXI OUT port for daisy chaining to another VXI-8340 module (dual port modules only)
- Eight front panel LEDs
  - **FAIL** LED indicates that the VXI-8340 is driving SYSFAIL onto the backplane.
  - **ONLINE** LED indicates when the VXI-8340 series is online. When it is offline, you can safely power cycle the VXI chassis.
  - **Link 1** LED indicates that a communication path has been established with the device attached to MXI IN.
  - **Link 2** LED indicates that a communication path has been established with the device attached to MXI OUT.
  - **SYSF** LED indicates that the VMEbus SYSFAIL line is asserted.
  - **ACC** LED indicates that the VXI-8340 is being accessed as a slave device on the VXI bus.
  - **TX/RX** LEDs indicate that the VXI-8340 is receiving data through the MXI port when green and transmitting data through the MXI port when yellow.
- Three SMB connectors
  - External clock
  - Trigger output
  - Trigger input
- Request offline button for taking the VXI-8340 series offline.
- Reset button for resetting the PCI-VXI bridge interface.

Advanced Configuration Options

The default hardware configuration of the VXI-8340 series should be acceptable for most systems. Refer to Appendix C, *Advanced Hardware Configuration Settings*, for other configuration options.

Software Description

The NI-VISA/NI-VXI bus interface software includes a Resource Manager, an interactive configuration and troubleshooting program, a comprehensive library of software routines for VXI/VME programming, a logging utility you can use for debugging, and graphical interactive control programs for interacting with VISA. You can use this software to seamlessly program multiple-mainframe configurations and have software compatibility across a variety of controller platforms.

NI-VISA has a comprehensive library of software routines not only for VXI/VME programming, but also for GPIB, GPIB-VXI, PXI, TCP/IP, and Serial. You can use this software to program instruments connected via different types of interfaces.

Measurement & Automation Explorer (MAX) helps merge NI-VXI and NI-VISA closer together. You use MAX to view your entire T&M system and configure various components, whether they are VXI, GPIB, PXI, or Serial devices. You can also add VME devices to your system easily with MAX and view them on a screen display along with the rest of your system.

MAX also features various options for running Resman. You can still execute Resman independently to configure your instruments after a power cycle. But you can also perform resource manager operations directly from MAX or configure it to run Resman automatically at startup.

The NI Spy utility tracks the calls your application makes to National Instruments T&M drivers, including NI-VXI, NI-VISA, and NI-488.2. NI Spy helps you debug your application by clearly highlighting the functions that return errors. You can let NI Spy keep a log of your program's calls to these drivers so that you can check them for errors at your convenience.

Software Configurations

There are two software configurations described in this manual:

- NI-VXI for Windows—This is a fully 32-bit native Plug and Play driver for Windows. You can run *only* 32-bit applications with this driver.
- NI-VISA for Windows—This is a 32-bit driver designed for Windows. Note that for VXI support, the NI-VXI driver must be installed. You can use this driver to develop and run 32-bit applications for Windows.

National Instruments Application Software

In addition to the NI-VISA/NI-VXI software, you can use the National Instruments LabVIEW and Measurement Studio application programs and instrument drivers to ease your programming task. These standardized programs match the modular virtual instrument capability of VXI and can reduce your VXI/VME software development time. These programs are fully VXI*plug&play* compliant and feature extensive libraries of VXI instrument drivers written to take full advantage of direct VXI control. LabVIEW and Measurement Studio include all the tools needed for instrument control, data acquisition, analysis, and presentation.

LabVIEW is an easy-to-use, graphical programming environment you can use to acquire data from thousands of different instruments, including IEEE 488.2 devices, VXI devices, serial devices, PLCs, and plug-in data acquisition boards. After you have acquired raw data, you can convert it into meaningful results using the powerful data analysis routines in LabVIEW. LabVIEW also comes with hundreds of instrument drivers, which dramatically reduce software development time, because you do not have to spend time programming the low-level control of each instrument.

Measurement Studio bundles LabWindows/CVI for C programmers, ComponentWorks for Microsoft Visual Basic programmers, and ComponentWorks++ for Microsoft Visual C++ programmers. Measurement Studio is designed for building measurement and automation applications with the programming environment of your choice:

- LabWindows/CVI is an interactive ANSI C programming environment designed for building virtual instrument applications. LabWindows/CVI delivers a drag-and-drop editor for building user interfaces, a complete ANSI C environment for building your test program logic, and a collection of automated code generation tools, as well as utilities for building automated test systems, monitoring applications, or laboratory experiments.
- ComponentWorks for Visual Basic is a collection of ActiveX controls designed for building virtual instrumentation systems. Based on ActiveX technology, ComponentWorks controls are configured through simple property pages. You can use the ComponentWorks GPIB, Serial, and VISA I/O controls and property pages to set up communication with your instruments.
- ComponentWorks++ for Visual C++ takes advantage of integrated C++ libraries and ActiveX to help you build measurement and automation applications. With the ComponentWorks++ instrument classes, you can use the IEEE 488.2 library and VISA, an

industry-standard I/O library, to communicate with GPIB, VXI, PXI, TCP/IP, or Serial devices using the same set of components.

If you want to use any of these application programs, install them before the NI-VISA/NI-VXI software installation. Both LabVIEW and Measurement Studio integrate the VXI and VISA libraries that are required to support your PCI-based MXI-3 products. You also get hundreds of complete instrument drivers, which are modular, source-code programs that handle the communication with your instrument to speed your application development.

2

Installation and Configuration

This chapter explains how to set up your test system.

Installing the Software

Use the Setup program that came with your NI-VISA/NI-VXI software to install the entire software package or a software update, or to reinstall software in the event that your files are accidentally erased.

Some of the utilities rely on the LabWindows/CVI Run-Time Engine. This software is installed, if necessary, during the NI-VISA/NI-VXI installation.

To be compliant with VXI*plug&play* specifications, a VXI controller must provide the VISA I/O driver library standardized by VXI*plug&play.* VISA ensures that your controller can run all VXI*plug&play*-compatible software now and in the future.

The NI-VISA software in this kit is compatible with the WINNT/GWINNT framework. With NI-VISA installed on your computer, you can run any VXI*plug&play* software that is compatible with this framework. This includes instrument drivers and executable soft front panel software that are included with VXI*plug&play*-compatible instruments from a variety of vendors.

Install the NI-VISA and NI-VXI Software

This section describes how to install the NI-VISA and NI-VXI software. Please read these directions carefully along with any messages on the screen before making your selections.

Note You can quit the Setup program at any time by pressing the **Cancel** button.

Setup is an interactive, self-guiding program that installs the NI-VISA and NI-VXI software and configures your system to use the software with the VXI-8340 series. Follow these steps to perform the installation:

1. Insert the CD-ROM labeled *NI-VXI v. 3.1 for Windows*.
2. Wait for Windows to launch the CD-ROM installation menu automatically. (If this does not happen, open the CD-ROM from Windows Explorer and double-click **Autorun**.)
3. Click on **Install NI-VXI Software for Windows**.
4. Click on the **Next** button at the **Welcome** screen to start the installation.

Note If you have a previous 32-bit (Windows 2000/NT/98) version of the NI-VXI software installed, Setup installs the new version over the previous version.

5. Select the destination for the installed files. The default is `C:\Program Files\National Instruments`.
6. Select the type of installation from the **Installation Options** screen:
   - **Typical** setup prompts you to make high-level choices of which driver(s) to install and the destination directories.
   - **Custom** setup gives you more control over which driver components you want installed on your system. This option is recommended for advanced users.
7. Follow the prompts. Confirm that you are ready to install and click **Next** to begin the installation.
8. Setup now copies the necessary files to your hard drive and creates program icons.

Note On some Windows systems, the OS will detect the MXI-3 interface as a "PCI Standard PCI-to-PCI Bridge," and Windows may ask you to specify the location of a driver information file (`.inf`) for the device. If this occurs, accept all of the default choices by clicking **Next** when prompted.

Complete the Software Installation

1. Review the information in any `readme` files that Setup prompts you to read.
2. When the installation process completes, restart the system for the changes to take effect.

Installing the Hardware

This section summarizes how to install your VXI-8340 series hardware. Your kit contains a VXI-8340 series interface module and may also contain a PCI-8330 series interface board and MXI-3 cable.

Caution To guard against electrostatic discharge, touch the antistatic plastic packages to a metal part of your computer or chassis before removing the boards from their packages. Your computer or chassis should be plugged in but powered off.

Figure 2-1 shows a system with a PCI bus-equipped computer and a VXI chassis with a VXI-8340 series module.

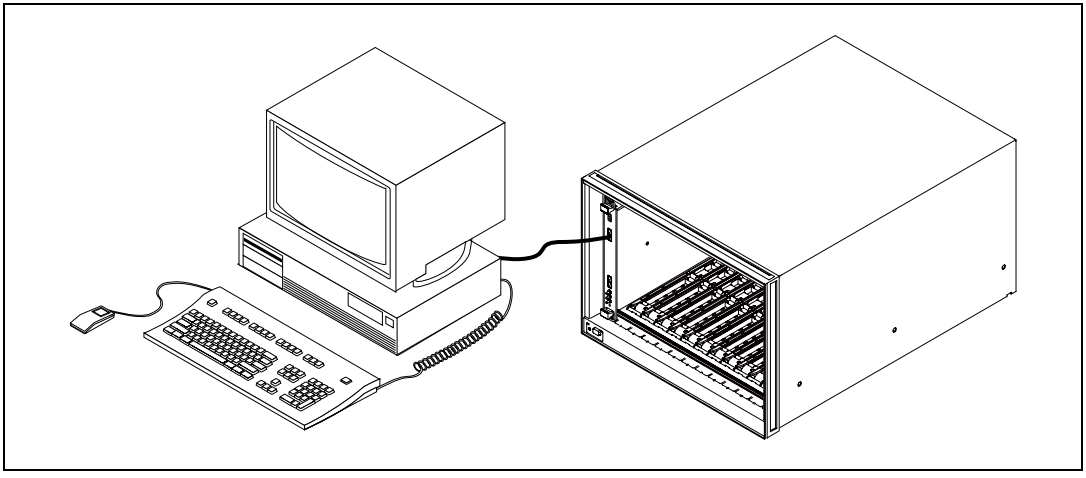


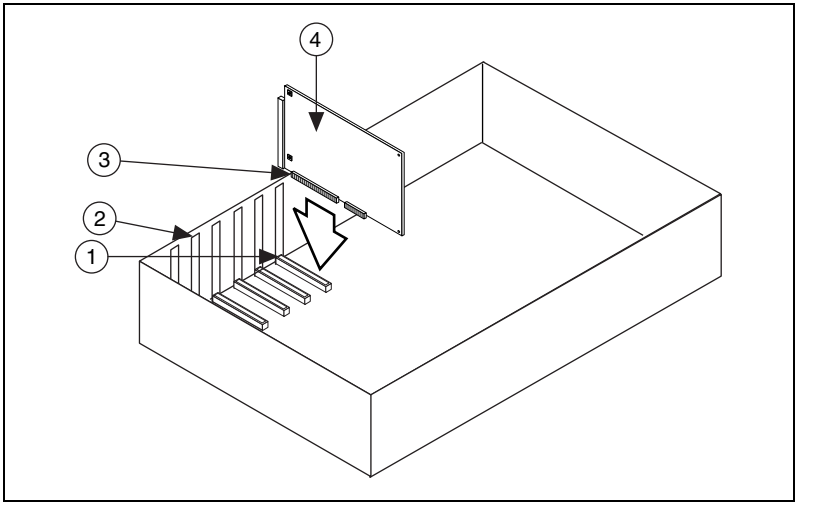
Figure 2-1. Typical VXI System with a VXI-8340 Series Interface Module

Install Your PCI-8330 Series Interface Board

If your kit includes a PCI-8330 series board, follow these general installation instructions:

1. Shut down and power off the computer.
2. Remove the computer chassis cover to expose the expansion slots and external access covers.
3. Select an available PCI slot in your computer. The slot you select should support bus mastering. Refer to your computer documentation to determine if the slot you select supports bus mastering.
4. Remove the corresponding expansion slot cover from the chassis.

5. Align the bus connector on the edge of the PCI-8330 series with the PCI bus slot.



1	PCI Bus Slot	3	Bus Connector
2	Expansion Slot Cut-Outs	4	PCI-8330 Series Board

Figure 2-2. Installing the PCI-8330 Series Interface Board

6. Carefully press the host adapter into the slot.
7. Secure the PCI-8330 series bracket to the computer chassis with the screw from the removed expansion slot cover.

Install Your VXI-8340 Series Interface Module

Install the VXI-8340 series in Slot 0 of your VXI chassis by doing the following:

1. Turn off the power.
2. Insert the VXI-8340 series into the chassis in Slot 0. See Figure 2-1.
3. Verify that all other VXI devices with system controller capability that are in the same chassis are *not* configured as system controller. The VXI-8340 series default configuration automatically detects whether it should be the VXI system controller.

Caution Having more than one device configured as system controller can damage the VXI system.

4. For VXI systems that include VME devices, ensure that the VME devices are not configured in the upper 16 KB (starting from 0xC000) of the A16 address space. This region is reserved for VXI device configuration registers, which are used for initializing, configuring, and interacting with VXI devices.

Note Also ensure that no other VXI devices in your system are configured for logical address 0. This is the default configuration for the VXI-8340 series.

Connect the Cable

Connect the MXI-3 cable to the PCI-8330 series and the MXI IN port of the VXI-8340 series, as shown in Figure 2-1.

Power on the System

1. Be sure the MXI-3 cable is connected securely.
2. Turn on the VXI chassis.

Note Always power on the VXI-8340 series at least 8 seconds before your PC. Refer to Appendix D, *Power Sequencing Requirements*, for more information regarding power-up requirements for the VXI-8340 series controllers.

3. Turn on the host computer.
4. If your system CMOS setup requires you to adjust PCI bus parameters, do so at this time.

Note The PCI bus usually assigns IRQs and port addresses automatically. However, in some PC systems, you may need to edit the PCI bus parameters manually in your CMOS setup. Some older PCs also require bus mastering to be enabled. Refer to your computer documentation for further instructions.

Software Configuration and Verification

Run MAX and Resman

1. Launch MAX from its desktop icon or **Start»Programs»National Instruments»Measurement & Automation Explorer**. MAX allows you to run Resman, the National Instruments Resource Manager. You must run Resman every time the chassis power is cycled so that your application can access devices in the VXI chassis. Refer to Appendix D, *Power Sequencing Requirements*, for more information regarding power-cycling requirements for the VXI-8340 series controllers.
2. After you run Resman, you are ready to use MAX to configure the National Instruments hardware in your system interactively. You can also configure MAX to run Resman automatically at every computer startup by selecting **Tools»NI-VXI»VXI Options**.
3. After you finish configuring the system through MAX, verify the system configuration through one of the interactive control utilities. Use VISAIC under NI-VISA or VIC under NI-VXI. You can also open VISA sessions using MAX by right-clicking on the device and selecting **Open VISA Session**.

For more details about the utilities in NI-VISA/NI-VXI, refer to Chapter 3, *Developing Your Application*.

Configuring the Hardware (Optional)

Because the default settings for your VXI-8340 series hardware are acceptable for most typical applications, this section is optional.

Refer to Appendix B, *Default Settings*, for a complete listing of the hardware and software default settings. Refer to Appendix C, *Advanced Hardware Configuration Settings*, if you want information about other possible settings.

Use MAX to change any configuration settings for the VXI-8340 series. For information on the software, including optional settings, use MAX and its online help. To access the help, select **Help»Help Topics**.

Using the Request Offline/Go Online Feature

The VXI-8340 series controllers use the National Instruments MXI-3 interface for a high-speed, software-transparent PCI link between a host controller and a VXI chassis. Because of the software-transparent nature of the MXI-3 connection, be careful when power-cycling the VXI chassis independently of the host controller.

To power-cycle a VXI chassis without rebooting the host computer, you must explicitly tell the NI-VXI software to take the VXI-8340 series system offline. When you take the VXI system offline, the NI-VXI driver saves important configuration information about the state of the VXI-8340 series, so that this state can be re-established when you restore power to the VXI chassis. Placing the system offline also prevents the NI-VXI driver from trying to communicate with the VXI-8340 series while the MXI-3 link is down. While offline, the VXI-8340 series is inaccessible to applications.

The ONLINE front-panel LED turns off when the VXI-8340 series is offline.

Note It is safe to remove power from a VXI chassis *only* after the VXI-8340 series is offline. Failure to take the controller offline before cycling power can lock up the system.

There are two ways to place a VXI-8340 series controller in the offline state. First, the VXI-8340 series includes a Request Offline button on the front panel. This button is useful if the VXI chassis is separated from the host controller by a great distance. Second, a **Request Offline** option is available in MAX by right-clicking on the VXI System icon under the **Devices and Interfaces** view.

To bring a VXI-8340 series controller back online, use the **Go Online** option in MAX by right-clicking on the VXI system icon under the **Devices and Interfaces** view. Choosing **Go Online** tells the NI-VXI software to restore the state of the VXI-8340 series to its pre-offline values.

The ONLINE front panel LED turns on when the VXI-8340 series is online. At this point, it is safe to access the VXI interface.

Note You cannot leave a VXI system offline across a system boot. You can bring a VXI system online only if that system was present when the host computer was last rebooted.

In general, follow these steps when taking a VXI-8340 series controller offline:

1. Request that the VXI-8340 series be taken offline:
   - Press the Request Offline button

   or

   - In MAX, right-click on the VXI system icon and choose **Request Offline**.
2. When the VXI-8340 series indicates it is offline by turning off the ONLINE LED, remove power from the VXI chassis. At this time, you can alter the VXI system configuration (replace devices, etc.).
3. Restore power to the VXI chassis.
4. In MAX, right-click on the VXI system and choose **Go Online**. When the system is ready for use, the VXI-8340 series turns on its ONLINE LED.
5. Run the Resource Manager in MAX.

When you use the Request Offline functionality, be sure there are no outstanding processes or sessions associated with the VXI-8340 series interface. If you attempt to use the Request Offline feature, and the VXI-8340 series fails to indicate the Offline condition, there may be open sessions to devices in the corresponding VXI chassis.

Also, be careful when VXI-8340 series controllers are connected in a daisy chain. To bring a daisy-chained VXI-8340 series offline, you must also take subordinate chassis in the chain offline. For example, if two VXI chassis are daisy chained, with chassis A ahead of chassis B in the chain, be sure chassis B is offline before taking chassis A offline.

3

Developing Your Application

This chapter discusses the software utilities you can use to start developing applications that use the NI-VXI/NI-VISA driver.

After installing the driver software, you can begin to develop your VXI or VISA application software. Be sure to check the release notes for the latest application development notes and changes.

Your software includes several utilities to assist you in your system development. These include MAX, Resman, NI Spy, VISAIC, and VIC. You can also access several examples to learn how to use NI-VISA or NI-VXI for certain tasks. Each component assists you with one of four steps of development: configuration, device interaction, programming, and debugging.

After installation, you can access these utilities through the Windows **Start** menu. Open the **National Instruments** or **VXIpnp** program group, then open either the **VXI** or **VXIpnp** subgroup to access the utilities you want to use.

Configuration

The configuration utilities in your kit are MAX and Resman. Resman performs VXI Resource Manager functions as described in the VXIbus specification. Its most important functions include configuring all devices on the VXI backplane for operation and allocating memory for devices that request it.

Note Power cycling resets all devices, so you need to run Resman to reconfigure them every time chassis power is cycled.

MAX presents a graphical display of your entire test and measurement system to help you configure various components. When you launch MAX, you see all your VXI, GPIB, GPIB-VXI, PXI, TCP/IP, and serial devices on the screen. You can view the properties (such as logical address, address space used, primary address, and so on) of each device by right-clicking on the device in the tree. When you view the properties of most National

Instruments devices, you can configure the hardware settings directly in the property pages.

MAX and Resman are designed to work together. You can run the Resource Manager through MAX by either clicking on the **Run VXI Resource Manager** button on the toolbar or selecting **NI-VXI»VXI Resource Manager** from the **Tools** menu. From the **VXI Options** dialog in the **Tools»NI-VXI** menu, you can also configure MAX to run Resman automatically when the computer boots up. Resman reports all errors it finds in your system to MAX. When you view your system through MAX, you can easily spot any errors in your system that Resman found.

You can find more information about MAX by using its online help. From MAX, open the **Help** menu and select **Help Topics**.

Device Interaction

After Resman has detected and configured all VXI/VME devices, you can view specific information on each device in your system by using the MAX utility. This utility includes a configuration view, which contains a description for each device, including each VXI device's logical address.

You can also interact with your VXI/VME devices by using the VIC or VISAIC utility (VIC for NI-VXI or VISAIC for NI-VISA). You can use these utilities to interactively control your VXI/VME devices without using a conventional programming language, LabVIEW, or Measurement Studio.

Note The VIC utility is not part of the basic NI-VXI installation. However, if you want to use VIC, run the NI-VXI installer and select a "custom" installation instead of "typical." In the **Select Features** dialog box, enable the **NI-VXI API Development** option and continue as prompted. After you reboot, VIC will be ready for use.

Note You can launch VIC or VISAIC from the **Tools** menu in MAX.

Try the following in VIC. In the **Command** entry field, type:

```
help vxiinreg
```

This help file shows you the syntax for this command, which reads VXI device configuration registers. The first argument is a logical address, and the second is the offset of the VXI device configuration register to be read.

Type:

```
vxiinreg 0,0
```

The **History** window shows the result of the command execution, such as:

```
Return Status (0): Success.

Value = 0x4ff6
```

If the value ends with `ff6`, you have successfully read the National Instruments manufacturer ID from the VXI/VME-MXI-3 ID register.

You may now want to read the configuration registers from other VXI devices in your system using the command `vxiinreg`. This command accesses only the upper 16 KB of A16 space. Try reading a register from each device listed in the MAX configuration view. In this way, you can verify that your VXI-8340 series can access each device in your VXI system successfully.

You can also access VXI and VME devices configured in A16, A24, and A32 address space by using the `vxiin` or `vxiout` commands. For more information regarding VIC operation and commands, refer to the VIC online help.

Alternately, you can use VISAIC to interact with your devices. VISAIC lists the available devices, similar to what MAX displays. By double-clicking on a given device, you can open a VISA session and access the device through it. For more information regarding VISAIC, use the right-click help available from all panels.

Programming with VXI

National Instruments provides two different programming interfaces for accessing your instruments: NI-VISA and NI-VXI. NI-VISA is the National Instruments implementation of the VISA API as defined by the VXI*plug&play* standard. It is very useful in situations where you have different types of instruments in your system (such as VXI, GPIB, PXI, and serial devices) because the NI-VISA functions have the same interface.

NI-VXI is the National Instruments proprietary interface for programming VXI instruments. Both NI-VXI and NI-VISA grant you register-level access of VXI instruments as well as messaging capability to message-based devices. With either interface you can service asynchronous events, such as triggers and signals, and also assert them.

The best way to learn how to program with NI-VXI or NI-VISA is by reviewing the example programs included in your software. In the examples directory are examples for many different types of applications.

If you are just getting started, you should first learn how to access registers with high-level calls and send messages with word serial functions. The NI-VISA examples of these tasks are called `VISAhigh.c` and `VISAws.c`. The NI-VXI examples are called `VXIhigh.c` and `VXIws.c`. You should use the other examples as you try more advanced techniques. Consult the *NI-VISA User Manual* or NI-VXI online help for additional information on these topics.

Note The NI-VXI development utilities are not part of the basic NI-VXI installation. However, if you want to use the NI-VXI development utilities, run the NI-VXI installer and select a "custom" installation instead of "typical." In the **Select Features** dialog box, enable the **NI-VXI API Development** option and continue as prompted. After you reboot, VIC will be ready for use.

Note The *NI-VISA User Manual* is in the `VXIpnp\os\NIvisa\manuals` directory, where `os` is your version of Windows. Use the Acrobat Reader program to open and navigate this manual.

Table 3-1 summarizes the topics addressed by the example programs.

Table 3-1. NI-VISA/NI-VXI Examples

Coverage	NI-VISA Example	NI-VXI Example
Message-Based Access	`VISAws.c`	`VXIws.c`
High-Level Register Access	`VISAhigh.c`	`VXIhigh.c`
Low-Level Register Access	`VISAlow.c`	`VXIlow.c`
Sharing Memory	`VISAmem.c`	`VXImem.c`
Interrupt Handling	`VISAint.c`	`VXIint.c`
Trigger Handling	`VISAtrig.c`	`VXItrig.c`

Note MAX includes special settings that you must use for low-level functions and memory sharing. Consult the MAX online help for information on setting these up.

Compiler Symbols for NI-VXI

You may need to define some symbols so that the NI-VXI library can work properly with your program.

Note Skip this section if you are programming with NI-VISA only. NI-VISA neither requires nor uses these symbols.

You can define the symbols using `#define` statements in the source code or you can use either the `/D` or `-D` option in your compiler (both the Microsoft and Borland compilers support the `/D` and `-D` options). If you use `#define` statements, you must define the symbols before including the NI-VXI header file `nivxi.h`. If you use the makefiles to compile the sample program, the makefile already defines the necessary symbols.

The following symbol is usually required. You must define it when using the Microsoft C or Borland C compiler.

- `VXINT` designates the application as a Windows 2000/NT/Me/98 application.

Note LabWindows/CVI automatically defines the correct symbol. You do not need to define `VXINT` when using LabWindows/CVI.

The `BINARY_COMPATIBLE` is optional. This symbol makes the application binary compatible with embedded VXI controllers, such as the National Instruments VXIpc series of embedded controllers. This option may cause a slight performance degradation when you use low-level VXIbus access functions.

If you define these symbols in your source code, your source code should look something like the following sample code:

```
#define VXINT
#define BINARY_COMPATIBLE
.
.
.
#include <nivxi.h>
```

If you define these symbols using the `/D` or `-D` compiler options, you should specify the following when invoking the compiler.

For the Microsoft C compiler:

```
/DVXINT /DBINARY_COMPATIBLE
```

For the Borland C compiler:

```
-DVXINT; BINARY_COMPATIBLE;
```

You also need to link in the appropriate import library for your code. If you are using a Microsoft C compiler, use the `nivxint.lib` in the `Program Files\National Instruments\VXI\NIVXI API\Win32\msc\` directory. If you are using a Borland C compiler, use the `nivxint.lib` in the `...\win32\borlandc\` directory.

Refer to the documentation that came with your compiler package for detailed instructions about using the compiler and the various tools (linker, debugger, and so on). Your compiler documentation is an important and useful source of information for writing, compiling, and debugging C programs.

Window Mapping

The MITE architecture allows a lot of flexibility in low-level mapping of VXI address spaces. In particular, the CPU interface of the MITE has windows that can be dynamically resized and relocated from CPU space to VXI space. The low-level functions have extensions that reflect this feature. Refer to the NI-VXI online help for information about these functions in NI-VXI. The NI-VISA online help and the *NI-VISA Programmer Reference Manual* cover this information for NI-VISA applications. As mentioned earlier, you can use the Acrobat Reader 3.0 to view and navigate through this manual.

The functions `MapVXIAddress()` and `viMapAddress()` check whether a window that can be shared already maps to the desired address space and location. If so, they return a pointer to that window. If the desired space is not already mapped, they set up a new MITE window to the VXI address and return a pointer to the new window.

The `MapVXIAddressSize()` function is the standard mechanism for specifying how large a window the driver should map on a call to `MapVXIAddress()`. The default size of a mapped window when using NI-VXI is 64 KB. In VISA, you specify the window size directly in `viMapAddress()`.

The success of this allocation depends on the availability of three factors:

- Address space in the User Window
- Number of MITE windows
- Memory for allocating data structures for the map

Address Space

The VXI-8340 series can decode any 32-bit address on the PCI bus as a VXI cycle, giving 4 GB of addressability, which can be used for windows on the VXI-8340 series. The operating system or computer architecture may limit which addresses can be assigned to the VXI-8340 series.

To change the address space, edit the **User window size** field in the **PCI** tab of the VXI-8340 series **Hardware Configuration** settings in MAX. This setting limits the total amount of memory you can map with `MapVXIAddress()` or `viMapAddress()`. If the User Window is disabled, the `MapVXIAddress()` function returns `NO_HARDWARE_SUPPORT (-1)`.

The NI-VXI online help implies that the error code `MAP_TIMEOUT (-8)` is returned when the window is in use. Because the MITE-based products have multiple hardware windows of variable size, the meaning of this error has been modified. `MapVXIAddress()` now returns the error code `MAP_TIMEOUT (-8)` whenever there are not enough resources to map the window.

For example, if you use `MapVXIAddressSize()` and `MapVXIAddress()` to request a 1 MB window to A32 space, and you request a User Window in MAX of only 64 KB, `MapVXIAddress()` returns the error code `MAP_TIMEOUT` because there are not enough resources to complete the request.

Number of MITE Windows

The MITE has eight CPU windows. NI-VXI uses four of these windows, leaving four for user applications.

Memory for Allocating Data Structures

You need to have sufficient memory available for the operating system to set up its necessary page tables. If you request a very large user window—hundreds of megabytes, for example—you may run out of memory.

MITE DMA

The MITE has two DMA channels to improve the throughput of block transfers to and from the VXI system. The DMA channels can use various high-speed bus protocols, such as the following:

- Burst mode (on the PCI bus)
- VME64 (on the VXI bus)

The DMA channels can transfer data between a VXI device and local memory, or between VXI devices. The DMA channel can handle contiguous or noncontiguous local memory. If it is handling noncontiguous memory, it can perform scatter-gather operations on the noncontiguous memory.

The `VXImove()` and `viMove/XX()` functions automatically use appropriate bus protocols and transfer types to efficiently perform the data transfer specified in the function. You can also use configuration options in MAX to instruct the NI-VXI/VISA software to use DMA channels for particular types of operations and to designate what protocols the channel should use. In addition, you can programmatically control which protocols to use in NI-VXI. See the NI-VXI online help for complete descriptions of `VXImove()` and other high-level functions. Notice that previously written NI-VXI and NI-VISA code uses the DMA capabilities of the MITE without modification.

To take full advantage of the throughput of the DMA channels, you should perform 32-bit transfers where both the source and the destination are longword aligned. If you need to transfer character data between devices of different byte orders—for example, between a big-endian device and an Intel 80*x*86-based Windows NT PC—transfer the data as longwords but adjust the byte-ordering parameters in `VXImove()` to get the correct data in the most efficient manner.

NI-VXI examples:

```
/* Transferring 32-bit data to a big-endian A32 device */
VXImove(0x0, userBuffer, 0x3, deviceOffset,
numDataPoints, 4);
/* Transferring 8-bit data to a big-endian A32 device */
VXImove(0x80, userBuffer, 0x3, deviceOffset,
numDataPoints / 4, 4);
```

Shared Memory

In the **Hardware Configuration** settings of the VXI-8340 series in MAX, you can share memory on your computer or from DRAM added to the VXI-8340 series. Right-click on any of the settings or consult the online help in MAX for more information. You can access shared memory on your computer using `VXImemAlloc()` in NI-VXI and `viMemAlloc()` in VISA.

Enhancements to the NI-VXI Software

The following sections describe the additional options beyond what is documented in the NI-VXI online help.

Compatibility

NI-VXI applications that follow the guidelines documented in the NI-VXI online help will work with NI-VXI for the VXI-8340 series.

System Configuration Functions

The `InitVXIlibrary()` function has a new return value of `INIT_RET_OK_RMERROR (2)`. If this value is returned, it means the NI-VXI library successfully initialized, but the Resource Manager has not been run successfully. Always run the Resource Manager before using the NI-VXI library.

Low-Level VXIbus Access Functions

Do not make any assumptions about the size and features of a window returned from `MapVXIAddress()`. You should use `GetWindowRange()` to determine the size of a window. The 32-bit value returned from `GetContext()` and passed to `SetContext()` has a new format. Applications that set the context bits directly for use in `SetContext()` may not be compatible with the new format for context. Because the MITE allows more flexible window mapping, extra bits have been added to this field to reflect these new features. Do not manipulate the context bits directly.

High-Level VXIbus Access Functions

For best performance, keep the following in mind when using `VXImove()`:

- Make sure your buffers are 32-bit aligned.
- Transfer 32-bit data whenever possible.
- Using VXI block access privileges significantly improves performance to devices that are capable of accepting block transfers.
- `VXImove()` must lock the user buffer in memory on virtual memory systems, so locking the buffer yourself optimizes `VXImove()`.
- Because `VXImove()` must build a scatter-gather list for the user buffer on paged memory systems, using a contiguous buffer optimizes `VXImove()`.

`VXImemAlloc()` returns 32-bit aligned, page-locked, contiguous buffers, which work efficiently with `VXImove()`, but only if the function returns `MEM_OK (0)`. A status of `MEM_OK_USE_MEMCOPY (1)` means this buffer cannot be used directly with `VXImove()`.

`VXImove()` can also move blocks of data to and from a single VXI address. This is commonly referred to as *FIFO mode*. For more information, refer to the NI-VXI online help.

Local Resource Access Functions

`VXImemAlloc()` does not allocate onboard RAM on the VXI-8340 series; it only allocates system RAM on the motherboard. If you want to access onboard RAM on the VXI-8340 series, access it as if it were VXI memory—that is, by using high-level or low-level VXIbus access functions. You can use `GetDevInfo()` on the VXI-8340 series device to determine the VXI address space and VXI address of this onboard RAM.

Debugging

NI Spy and VISAIC are useful utilities that can aid in identifying the causes of problems in your application.

NI Spy tracks the calls your application makes to National Instruments T&M drivers including NI-VXI, NI-VISA, and NI-488.2. NI-488.2 users may notice that NI Spy is very similar to GPIB Spy. It highlights functions that return errors, so you can quickly spot which functions failed during your development. NI Spy can log the calls your program makes to these drivers so you can check them for errors at your convenience.

You can also control your instruments interactively using VISAIC. You can use VISAIC to control and communicate with your instruments with NI-VISA without having to write a program. This utility is an excellent platform for quickly testing instruments and learning how to communicate with them.

Refer to the online help for instructions on how to use VISAIC and to learn about their features. In VISAIC, you can right-click to reach **What's This** help and function help.

A

Specifications

This appendix lists the specifications for the VXI-8340 series and PCI-8330 series modules.

Requirements

VXIbus configuration space..... 64 B

A24 or A32 space..... Programmable

Default ..... None

Environmental

Temperature

Operating.....0 to 55 °C

Storage .....–20 to 70 °C

Relative humidity

Operating ..... 10% to 90% noncondensing

Storage ..... 5% to 95% noncondensing

EMI ..... FCC Class A verified, EC verified

Power Requirement

+5 V

Typical ..... 4.30 A

Maximum (fused) ..... 7 A

–5.2 V

Typical ..... 180 mA

Maximum (fused) ..... 1 A

–2 V

Typical ..... 110 mA

Maximum (fused) ..... 1 A

+12 V/–12 V

Typical.....0 A

Maximum (fused).....1 A

Physical

Dimensions

VXI-8340 series.....233.35 by 340 mm (9.187 by 13.386 in.)

PCI-8330 series .....107 by 175 mm (4.2 by 6.9 in.)

Slot requirements

VXI-8340 series.....Single VXI C-size slot

PCI-8330 series .....PCI bus slot

Compatibility .....Fully compatible with VXI specification

VXI keying class .....Class 1 TTL

MTBF .....Contact factory

VMEbus Capability Codes

A32, A24, A16 (master)
VMEbus master A32, A24, and A16 addressing

A32, A24, A16 (slave)
VMEbus slave A32, A24, and A16 addressing

D64, D32, D16, D08(EO) (master)
VMEbus master D64, D32, D16, and D08 data sizes

D64, D32, D16, D08(EO) (slave)
VMEbus slave D64, D32, D16, and D08 data sizes

BLT, MBLT (master)
VMEbus master block and D64 transfers

BLT, MBLT (slave)
VMEbus slave block and D64 transfers

RMW (master)
VMEbus master read/modify/write transfers

RMW (slave)
VMEbus slave read/modify/write transfers

RETRY (master)
VMEbus master retry support

RETRY (slave)
VMEbus slave retry support

FSD
First slot detector

SCON
VMEbus System Controller (Automatic Detection)

PRI, RRS
Prioritized or Round Robin Select arbiter

ROR, FAIR
Release on Request and FAIR bus requester

IH(7–1)
Interrupt handler for levels 7–1

I(7–1)
Interrupt requester for levels 7–1

D32, D16, D08(O) (Interrupt Handler)
VMEbus D32, D16, D08(O) interrupt handler

D32, D16, D08(O) (Interrupter)
VMEbus D32, D16, D08(O) interrupter

ROAK, RORA
Release on Acknowledge or Register Access interrupter

BTO(x)
VMEbus bus timer (programmable limit)

LOCK
Can lock the VMEbus for indivisible transfers

B

Default Settings

This appendix summarizes the default settings for the hardware and software in the VXI-8340 kit. If you need more information about a particular setting or want to try a different configuration, refer to Appendix C, *Advanced Hardware Configuration Settings*, for your hardware reference and the MAX online help for your software reference.

Note There are no hardware settings on the PCI-8330 series boards.

Hardware Settings

Table B-1. VXI-8340 Series Hardware Default Settings

Hardware Component	Default Setting
J1—VXIbus Slot 0/Non-Slot 0	Automatic detection
S2—SMB CLK10 direction	OUT: drive onboard CLK10 onto SMB connector and backplane
S1—Configuration EEPROM	Do *not* load from factory setting
DRAM SIMMs installed	Per customer order

Software Settings

Table B-2. MAX Device Tab Default Settings

Editor Field	Default Setting
Logical address	0
Device class	Message based
Size of Servant area	0
System interrupt level	Disabled

Table B-2. MAX Device Tab Default Settings (Continued)

Editor Field	Default Setting
Number of handlers	1
Number of interrupters	0

Table B-3. MAX Shared Memory Tab Default Settings

Editor Field	Default Setting
Memory sharing	Don't share memory
Shared RAM size	None
Reserved physical memory	0
Lower half window byte swapping	Disabled
Upper half window byte swapping	Disabled
Map upper and lower halves at same PCI address	Disabled

Table B-4. MAX VXI Bus Tab Default Settings

Editor Field	Default Setting
Bus timeout value	125 µs
VXI retry generation	Enabled
Automatic retries	Enabled
A24/A32 write posting	Enabled
Transfer limit	256
Requester mode	Release on Request
Request level	3
Fair requester	Enabled
Bus arbitration mode	Prioritized
Arbiter timeout	Enabled

Table B-5. MAX PCI Tab Default Settings

Editor Field	Default Setting
Enable low-level register access API support	Enabled
User window size	64 KB

Table B-6. MAX SMB Tab Default Settings

Editor Field	Default Setting
Enable 50 Ω termination for EXTCLK SMB connector	Disabled
Invert external CLK10 polarity	Disabled
Enable 50 Ω termination for TRIG IN	Disabled

C

Advanced Hardware Configuration Settings

This appendix describes the factory-default and alternate hardware configuration settings of the VXI-8340 series module. The module is set at the factory for the most commonly used configuration. Use this appendix if you want to try a different hardware configuration, or if you would like more information on a particular setting. This information is intended for more advanced users.

Note There are no hardware settings on the PCI-8330 series boards.

Hardware Default Settings

The following hardware configuration settings are user configurable:

- VXIbus Slot 0/Non-Slot 0
- Configuration EEPROM
- VXIbus CLK10 routing

Figure C-1 shows the factory-default settings of the user-configurable jumper and switches on the VXI-8340 series. The slot selection jumper and switches are at the top of the board.

Note Do *not* attempt an alternate setting unless you are familiar with its purpose. In addition, do not reconfigure any switches or jumpers not described in this appendix unless directed by National Instruments support.

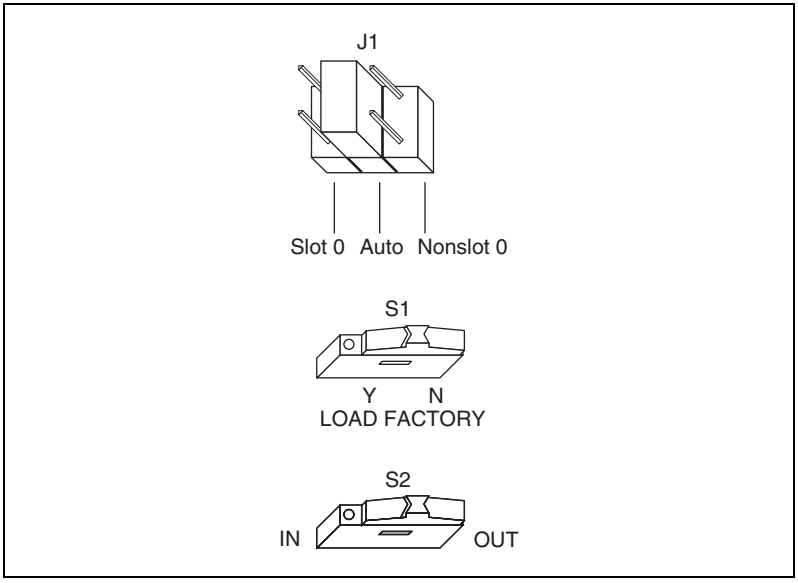


Figure C-1. VXI-8340 Series Default Configuration Settings

VXIbus Slot 0/Non-Slot 0

The VXI-8340 series is configured at the factory to automatically detect if it is installed in Slot 0 of a VXIbus mainframe.

You can manually configure the VXI-8340 series for either Slot 0 or Non-Slot 0 operation by defeating the automatic-detection circuitry. Use the three-position jumper J1 to select automatic Slot 0 detection, Slot 0, or Non-Slot 0 operation. Figure C-2 shows these three settings.

Caution Do *not* install a device configured for Slot 0 into another slot without first reconfiguring it to either Non-Slot 0 or automatic configuration. Neglecting to do this could damage the device, the VXIbus backplane, or both.

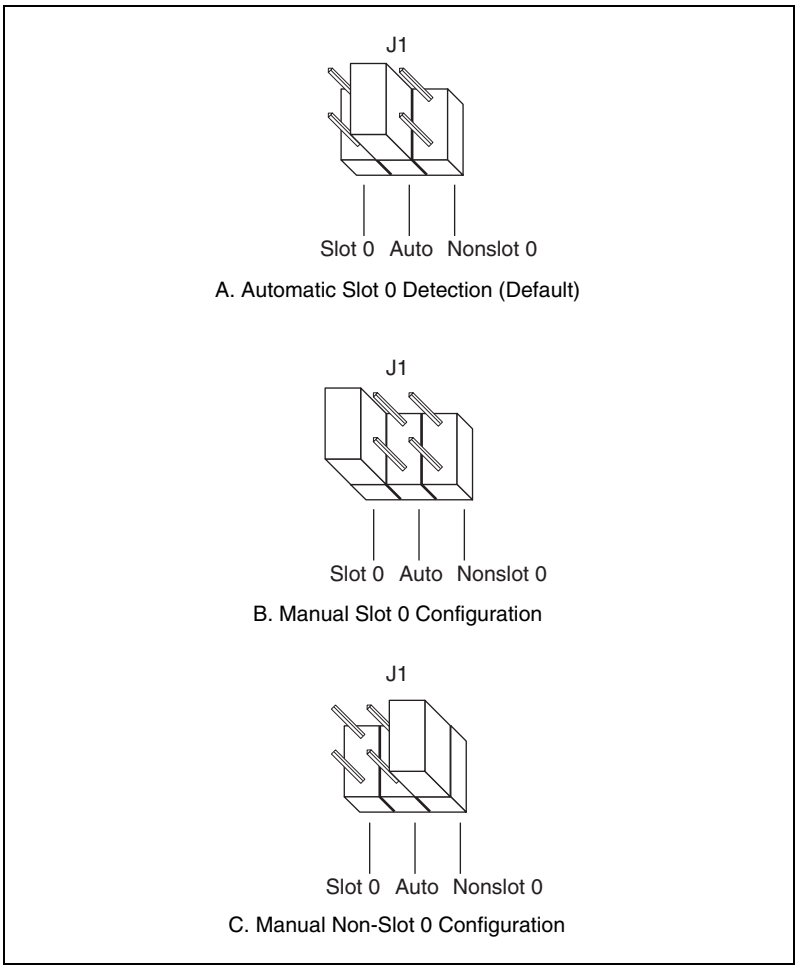


Figure C-2. VXIbus Slot Configuration

When the VXI-8340 series is installed in Slot 0, it becomes the VXIbus System Controller. In this role, it has VXIbus Data Transfer Bus Arbiter circuitry that accepts bus requests on all four VXIbus request levels, prioritizes the requests, and grants the bus to the highest priority requester. As VXIbus System Controller, the VXI-8340 series also uses an onboard 16 MHz oscillator to drive the 16 MHz VXIbus system clock.

As required by the VXIbus specification, the VXI-8340 series drives the 10 MHz signal CLK10 on a differential ECL output when installed in Slot 0. When not installed in Slot 0, the VXI-8340 series only receives the CLK10 signal.

Configuration EEPROM

The VXI-8340 series has an onboard EEPROM, which stores register values that are loaded at power-on. The EEPROM is divided into two halves—a factory-configuration half, and a user-configuration half. Both halves were factory configured with the same values. You can modify the user-configurable half, while the factory-configured half stores a back-up of the default settings.

The Load Factory switch (switch S1) causes the VXI-8340 series to boot off the factory-configured half instead of the user-modified settings. This is useful in the event that the user-configured half of the EEPROM becomes corrupted in such a way that the VXI-8340 series powers up in an unusable state.

Figure C-3 shows the configuration settings for EEPROM operation.

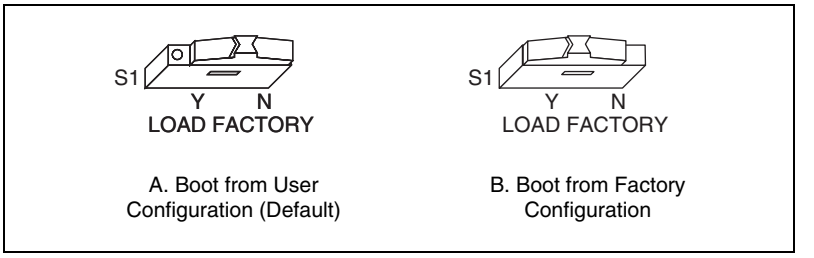


Figure C-3. EEPROM Operation

VXIbus CLK10 Routing

When the VXI-8340 series controller is installed in Slot 0 of your VXI mainframe, it must drive the VXIbus CLK10 signal. You can drive this signal from one of two sources: an onboard 10 MHz oscillator or an external TTL clock generator attached to the front panel EXT CLK connector. The CLK10 source is determined by the position of hardware switch S2.

Figure C-4 shows the configuration settings for CLK10 signal routing.

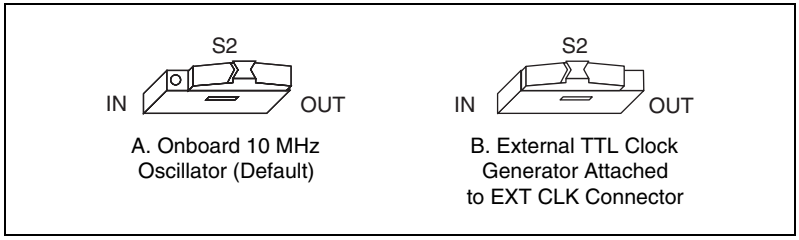


Figure C-4. CLK10 Signal Routing

It is important to note that when the EXT CLK connector is not being used as an input to drive the VXI CLK10 signal (that is, S2 is set to OUT), it serves as an output onto which the VXI-8340 series drives the CLK10 signal.

Caution If S2 is set to OUT, do *not* attach any device that will drive a signal to the EXT CLK connector. Doing so may damage the device or your VXI-8340 series module.

D

Power Sequencing Requirements

The VXI-8340 series uses MXI-3 technology to extend the PCI bus of a host computer to a VXI chassis transparently, so your host computer can function as an embedded CPU module. Because of this transparent PCI extension, it is important to maintain a strict power-on sequence when turning on VXI chassis in a VXI-8340 system.

For the system software to discover and configure a VXI-8340 series controller, power on all VXI-8340 chassis in the system before enabling power on the host computer. If any chassis is not present when the host computer initiates its boot process, the host computer may not configure that VXI controller. If a VXI system is absent when the system boots, the host must be rebooted to discover the missing VXI system.

Note You must power on the VXI-8340 series at least 8 seconds before your PC.

When daisy-chaining VXI-8340 series controllers, power on VXI chassis in reverse order, starting with the chassis furthest from the host computer. For example, if Chassis A is connected to the host computer, and Chassis B is daisy-chained to Chassis A, power on Chassis B before Chassis A, and power on Chassis A before the host computer.

These power sequencing requirements also apply to the Request Offline/Go Online feature (see the *Using the Request Offline/Go Online Feature* section in Chapter 2, *Installation and Configuration*). If multiple chassis in a daisy chain are offline, power on these chassis and bring them back online starting with the chassis furthest from the host computer. Using the example above, if you take Chassis A offline, you must also take Chassis B offline. When you bring these VXI systems back online, power on Chassis B first, followed by Chassis A.

E

Common Questions

This appendix addresses common questions you may have about using the NI-VISA/NI-VXI software with the VXI-8340 series.

I need more devices than can fit in one chassis. How can I expand my VXI-8340 series system?

There are two ways to expand your system beyond a single chassis:

- You can add additional VXI-8340 series controllers, either by daisy-chaining through a dual-port VXI-8340 or by adding another single-port VXI-8340 kit (PCI-8330 and single-port VXI-8340).
- If you need all your VXI devices to appear as if they are part of the same VXI system (that is, they share an address space, interrupts, and triggers), VXI-MXI-2 allows you to add more chassis to your system. VXI-MXI-2 extends full VXI functionality across multiple mainframes in accordance with the VXI-6 specification. Place a VXI-MXI-2 in the same frame as the Slot 0 VXI-8340 controller, and another VXI-MXI-2 in Slot 0 of the mainframe you want to add to the system. You can then fill up this mainframe with additional devices.

How can I determine the serial number and hardware revision of the VXI-8340 series module?

Run MAX and right-click on the name of the VXI-8340 series module. Select **Hardware Configuration**, and the dialog box for the VXI-8340 series module is displayed. The title bar includes the serial number and hardware revision of the module.

What is Resman?

Resman is the utility that performs the duties of a VXI Resource Manager as discussed in the VXIbus specification. When you set a National Instruments controller to Logical Address 0, you will at some point need to run Resman to configure your VXI instruments. If your controller uses a different (non-zero) logical address and is a message-based device, you need to start Resman before running it on the Logical Address 0 computer.

When do I need to run Resman?

Run Resman whenever you need to configure your VXI instruments. For example, if you power-cycle your VXI chassis, your instruments will be reset, and you will need to run Resman to configure them. You can get into trouble if you run Resman when your devices are not in a reset state. Therefore, if you have to run Resman after running it once, you should reset all of your VXI instruments.

You can perform resource manager operations from within MAX after you install the VXI-8340 series and cable it to your computer. With the VXI-8340 series, you may need to run the Resman utility if you boot your computer before turning on your VXI chassis or if you power-cycle your VXI chassis while the external PC remains on. In these cases, the instruments would have been reset without the computer rebooting. You will need to run the Resman utility or configure your system in MAX to initialize your VXI system.

Which NI-VXI utility program must I use to configure the VXI-8340 series?

Use MAX to configure the VXI-8340 series.

Which NI-VXI utility program must I use to initialize the VXI-8340 series?

The VXI-8340 series initializes itself at chassis power-up.

How do I handle VME devices?

Although there is no way to automatically detect VME devices in a system, you can add them easily through the **Add Device Wizard** in MAX. Through this procedure, you can reserve resources for each of your VME devices and configure MAX to show VME devices in the configuration tree with all your other devices.

Which NI-VXI utility program must I use to perform startup Resource Manager operations?

Use the Resman program to perform startup Resource Manager operations. Resman uses the settings configured in MAX.

You can also run Resource Manager operations from MAX. Through MAX, you can also configure Resman to run automatically at computer startup.

What can I do to make sure that my system is up and running?

The fastest method for testing the system is to run Resman. This program attempts to access memory in the upper A16 address space of each device in the system. If Resman does not report any problems, the VXI-8340 series communication system is operational.

To test individual devices, you can use the VIC or VISAIC program to interactively issue NI-VXI functions or NI-VISA operations, respectively. You can use the `VXIin()` and `VXIout()` functions or the `VXIinReg()` and `VXIoutReg()` functions to test register-based devices by programming their registers. If you have any message-based devices, you can send and receive messages with the `WSwrt()` and `WSrd()` functions. Notice that `VXIinReg()` and `VXIoutReg()` are for VXI devices only, but you can use `VXIin()` and `VXIout()` for both VXI and VME.

Finally, if you are using LabVIEW or LabWindows/CVI and you have instrument drivers for the devices in your chassis, you can use the interactive features of these programs to quickly test the functionality of the devices.

What should I do if I get a `Configuration EEPROM is Invalid` message?

There are several reasons why you might get the **`Configuration EEPROM is Invalid`** message. If you turned off the VXI chassis while the configuration update process was still in progress, the module functions normally except when running MAX. To correct these problems, reboot the VXI chassis with the Load Factory switch set (as described in Appendix C, *Advanced Hardware Configuration Settings*) and update the configuration, or load the configuration from file.

What do the LEDs on the front of the VXI-8340 series mean?

- The **FAIL** LED indicates that the VXI-8340 is driving SYSFAIL onto the backplane.
- The **ONLINE** LED indicates when the VXI-8340 series is online. When it is offline, you can safely power cycle the VXI chassis.
- The **Link 1** LED indicates that a communication path has been established with the device attached to MXI IN.
- The **Link 2** LED indicates that a communication path has been established with the device attached to MXI OUT.
- The **SYSF** LED indicates that the VMEbus SYSFAIL line is asserted.

- The **ACC** LED indicates that the VXI-8340 is decoding a VXI cycle.
- The **TX/RX** LEDs indicate that the VXI-8340 is receiving data through the MXI port when green and transmitting data through the MXI port when yellow.

What kind of signal is CLK10 and what kind of signal do I need for an external CLK10?

CLK10 is a differential ECL signal on the VXIbus backplane. However, the oscillator for the VXI-8340 series and the EXT CLK input from the front panel use TTL. Therefore, supply a TTL-level signal for EXT CLK, and our voltage converters will convert the signal to differential ECL.

What are shared memory and dual-ported memory?

These terms refer to a block of memory that is accessible to both a client and a server. The memory block operates as a message buffer for communications. Shared memory is applicable only if you are using either A24 or A32 address space.

F

Technical Support Resources

Web Support

National Instruments Web support is your first stop for help in solving installation, configuration, and application problems and questions. Online problem-solving and diagnostic resources include frequently asked questions, knowledge bases, product-specific troubleshooting wizards, manuals, drivers, software updates, and more. Web support is available through the Technical Support section of `ni.com`.

NI Developer Zone

The NI Developer Zone at `ni.com/zone` is the essential resource for building measurement and automation systems. At the NI Developer Zone, you can easily access the latest example programs, system configurators, tutorials, technical news, as well as a community of developers ready to share their own techniques.

Customer Education

National Instruments provides a number of alternatives to satisfy your training needs, from self-paced tutorials, videos, and interactive CDs to instructor-led hands-on courses at locations around the world. Visit the Customer Education section of `ni.com` for online course schedules, syllabi, training centers, and class registration.

System Integration

If you have time constraints, limited in-house technical resources, or other dilemmas, you may prefer to employ consulting or system integration services. You can rely on the expertise available through our worldwide network of Alliance Program members. To find out more about our Alliance system integration solutions, visit the System Integration section of `ni.com`.

Worldwide Support

National Instruments has offices located around the world to help address your support needs. You can access our branch office Web sites from the Worldwide Offices section of `ni.com`. Branch office Web sites provide up-to-date contact information, support phone numbers, e-mail addresses, and current events.

If you have searched the technical support resources on our Web site and still cannot find the answers you need, contact your local office or National Instruments corporate. Phone numbers for our worldwide offices are listed at the front of this manual.

Glossary

Prefix	Meaning	Value
p-	pico-	10^{-12}
n-	nano-	10^{-9}
µ-	micro-	10^{-6}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}
G-	giga-	10^{9}
t-	tera-	10^{12}

Symbols

°	degrees
Ω	ohms

A

A	amperes
A16 space	VXIbus address space equivalent to the VME 64 KB *short* address space. In VXI, the upper 16 KB of A16 space is allocated for use by VXI devices' configuration registers. This 16 KB region is referred to as VXI configuration space.
A24 space	VXIbus address space equivalent to the VME 16 MB *standard* address space.
A32 space	VXIbus address space equivalent to the VME 4 GB *extended* address space.
address	Character code that identifies a specific location (or series of locations) in memory. In VISA, it identifies a resource.

address modifier	One of six signals in the VMEbus specification used by VMEbus masters to indicate the address space in which a data transfer is to take place.
address space	A set of 2^n memory locations differentiated from other such sets in VXI/VMEbus systems by six addressing lines known as address modifiers. *n* is the number of address lines required to uniquely specify a byte location in a given space. Valid numbers for *n* are 16, 24, and 32. In VME/VXI, because there are six address modifiers, there are 64 possible address spaces.
ANSI	American National Standards Institute
API	Application Programming Interface—the direct interface that an end user sees when creating an application.
arbitration	A process in which a potential bus master gains control over a particular bus.

B

b	Bit—one binary digit, either 0 or 1.
B	Byte—eight related bits of data, an 8-bit binary number. Also used to denote the amount of memory required to store one byte of data.
backplane	An assembly, typically a printed circuit board, with 96-pin connectors and signal paths that bus the connector pins. A C-size VXIbus system will have two sets of bused connectors called J1 and J2. A D-size VXIbus system will have three sets of bused connectors called J1, J2, and J3.
BERR*	bus error signal
BIOS	Basic Input/Output System. BIOS functions are the fundamental level of any PC or compatible computer. BIOS functions embody the basic operations needed for successful use of the computer's hardware resources.
block-mode transfer	An uninterrupted transfer of data elements in which the master sources only the first address at the beginning of the cycle. The slave is then responsible for incrementing the address on subsequent transfers so that the next element is transferred to or from the proper storage location. A VME data transfer may have no more than 256 elements.

bus	The group of conductors that interconnect individual circuitry in a computer. Typically, a bus is the expansion vehicle to which I/O or other devices are connected. Examples of buses include the ISA bus, PCI bus, VXI bus, and VME bus.
bus error	An error that signals failed access to an address. Bus errors occur with low-level accesses to memory and usually involve hardware with bus mapping capabilities. For example, nonexistent memory, a nonexistent register, or an incorrect device access can cause a bus error.
bus master	A device that is capable of requesting the Data Transfer Bus (DTB) for the purpose of accessing a slave device.
byte order	How bytes are arranged within a word or how words are arranged within a longword. Motorola ordering stores the most significant byte (MSB) or word first, followed by the least significant byte (LSB) or word. Intel ordering stores the LSB or word first, followed by the MSB or word.

C

C	Celsius
CLK10	A 10 MHz, ±100 ppm, individually buffered (to each module slot), differential ECL system clock that is sourced from Slot 0 of a VXIbus mainframe and distributed to Slots 1 through 12 on P2. It is distributed to each slot as a single-source, single-destination signal with a matched delay of under 8 ns.
CMOS	Complementary Metal Oxide Semiconductor—a process used in making chips.
Commander	A message-based device that is also a bus master and can control one or more Servants.
configuration registers	A set of registers through which the system can identify a module device type, model, manufacturer, address space, and memory requirements. To support automatic system and memory configuration, the VXI specification requires that all VXIbus devices have a set of such registers.

D

Data Transfer Bus	DTB; one of four buses on the VMEbus backplane. The DTB is used by a bus master to transfer binary data between itself and a slave device.
DMA	Direct Memory Access—a method by which data is transferred between devices and internal memory without intervention of the central processing unit. DMA is the fastest method of transferring data to/from computer memory.
DRAM	Dynamic RAM (Random Access Memory)—storage that the computer must refresh at frequent intervals.
dynamic configuration	A method of automatically assigning logical addresses to VXIbus devices at system startup or other configuration times.
dynamically configured device	A device that has its logical address assigned by the Resource Manager. A VXI device initially responds at Logical Address 255 when its MODID line is asserted. The Resource Manager subsequently assigns it a new logical address, to which the device responds until powered down.

E

ECL	Emitter-Coupled Logic
EEPROM	Electronically Erasable Programmable Read Only Memory—ROM that can be erased with an electrical signal and reprogrammed.
embedded controller	An intelligent CPU (controller) interface plugged directly into the VXI backplane, giving it direct access to the VXIbus. It must have all of its required VXI interface capabilities built in.
EMC	electromagnetic compliance
EMI	electromagnetic interference
external trigger	A voltage pulse from an external source that triggers an event.

F

fair requester	A VXIbus device that will not arbitrate for the VXIbus after releasing it until it detects the bus request signal inactive. This ensures that all requesting devices will be granted use of the bus.

G

g	(1) grams (2) a measure of acceleration equal to 9.8 m/s^2
GPIB	General Purpose Interface Bus (IEEE 488)
g_{RMS}	A measure of random vibration. The root mean square of acceleration levels in a random vibration test profile.

H

hex	Hexadecimal—the numbering system with base 16, using the digits 0 to 9 and letters A to F.
Hz	hertz; cycles per second

I

I/O	Input/output—the techniques, media, and devices used to achieve communication between machines and users.
IEC	International Electrotechnical Commission. The IEC publishes internationally recognized standards. IEC 60068 contains information on environmental testing procedures and severities.
IEEE	Institute of Electrical and Electronics Engineers
in.	inches
instrument driver	A set of routines designed to control a specific instrument or family of instruments, and any necessary related files for LabWindows/CVI or LabVIEW.

interrupt	A means for a device to request service from another device; a computer signal indicating that the CPU should suspend its current task to service a designated activity.
interrupt handler	A VMEbus functional module that detects interrupt requests generated by interrupters and responds to those requests by requesting status and identify information.
interrupt level	The relative priority at which a device can interrupt.
IRQ*	interrupt signal

K

K	kilo—(1) the standard metric prefix for 1,000, or 10^3, used with units of measure such as volts, hertz, and meters; (2) the prefix for 1,024, or 2^{10}, used with B (byte) in quantifying data or computer memory.

L

logical address	An 8-bit number that uniquely identifies each VXIbus device in a system. It defines the A16 register address of a device, and indicates Commander and Servant relationships.

M

m	meters
M	mega—(1) the standard metric prefix for 1 million or 10^6, when used with units of measure such as volts and hertz; (2) the prefix for 1,048,576, or 2^{20}, when used with B (byte) to quantify data or computer memory.
master	A functional part of a VME/VXIbus device that initiates data transfers on the backplane. A transfer can be either a read or a write.
message-based device	An intelligent device that implements the defined VXIbus registers and communication protocols. These devices are able to use Word Serial Protocol to communicate with one another through communication registers.

MODID	Module ID lines—used in VXI to geographically locate boards and to dynamically configure boards.
MTBF	Mean Time Between Failure
MXI-3	A PCI Master/Slave system implementing the PCI-to-PCI Bridge register set. It couples two physically separate PCI buses with either a copper or fiber optic data link capable of 1.5 Gbits/s serial data rates.

N

NI-488.2 or NI-488.2M	The National Instruments industry-standard software for controlling GPIB instruments.
NI-DAQ	The National Instruments industry-standard software for data acquisition instruments.
NI-VISA	The National Instruments implementation of the VISA standard; an interface-independent software that provides a unified programming interface for VXI, GPIB, and serial instruments.
NI-VXI	The National Instruments bus interface software for VME/VXIbus systems.
Non-Slot 0 device	A device configured for installation in any slot in a VXIbus mainframe other than Slot 0. Installing such a device into Slot 0 can damage the device, the VXIbus backplane, or both.

P

PCI	Peripheral Component Interconnect. The PCI bus is a high-performance 32-bit or 64-bit bus with multiplexed address and data lines.

R

register-based device	A Servant-only device that supports VXIbus configuration registers. Register-based devices are typically controlled by message-based devices via device-dependent register reads and writes.
Resman	The name of the National Instruments Resource Manager in NI-VXI bus interface software. *See* Resource Manager.

Resource Manager	A message-based Commander located at Logical Address 0, which provides configuration management services such as address map configuration, Commander and Servant mappings, and self-test and diagnostic management.
retry	An acknowledge by a destination that signifies that the cycle did not complete and should be repeated.
RMS	Root mean squared. *See* gRMS.

S

s	seconds
SIMM	Single In-line Memory Module
slave	A functional part of a VME/VXIbus device that detects data transfer cycles initiated by a VMEbus master and responds to the transfers when the address specifies one of the device's registers.
Slot 0 device	A device configured for installation in Slot 0 of a VXIbus mainframe. This device is unique in the VXIbus system in that it performs the VXI/VMEbus System Controller functions, including clock sourcing and arbitration for data transfers across the backplane. Installing such a device into any other slot can damage the device, the VXIbus backplane, or both.
SMB	Sub Miniature Type B connector that features a snap coupling for fast connection.
statically configured device	A device whose logical address cannot be set through software; that is, it is not dynamically configurable.
SYSFAIL	A VMEbus signal that is used by a device to indicate an internal failure. A failed device asserts this line. In VXI, a device that fails also clears its PASSed bit in its Status register.

T

trigger	Either TTL or ECL lines used for intermodule communication.
TTL	Transistor-Transistor Logic

V

V	volts
VIC	VXI Interactive Control program, a part of the NI-VXI bus interface software. Used to program VXI devices and develop and debug VXI application programs.
VISA	Virtual Instrument Software Architecture. This is the general name given to VISA and its associated architecture.
VISAIC	VISA Interactive Control program, a part of the NI-VISA software. Used to program devices and develop and debug application programs.
VITA	VMEbus International Trade Association
VME	Versa Module Eurocard or IEEE 1014
VMEbus System Controller	A device configured for installation in Slot 0 or a VXIbus mainframe or the first slot in a VMEbus chassis. This device is unique in the VMEbus system in that it performs the VMEbus System Controller functions, including clock sourcing and arbitration for data transfers across the backplane. Installing such a device into any other slot can damage the device, the VMEbus/VXIbus backplane, or both.
VXIbus	VMEbus Extensions for Instrumentation

W

W	watts
Word Serial Protocol	The simplest required communication protocol supported by message-based devices in a VXIbus system. It utilizes the A16 communication registers to transfer data using a simple polling handshake method.
write posting	A mechanism that signifies that a device will immediately give a successful acknowledge to a write transfer and place the transfer in a local buffer. The device can then independently complete the write cycle to the destination.

Index

Symbols

/D or -D compiler options, 3-5

A

ACC LED, 1-3, E-4
address space, 3-7
advanced hardware configuration settings, C-1
allocating data structures
    memory for, 3-7
application development, 3-1
    address space, 3-7
    compiler symbols for NI-VXI, 3-5
    configuration, 3-1
    debugging, 3-10
    device interaction, 3-2
    high-level VXIbus access functions, 3-9
    local resource access functions, 3-10
    low-level VXIbus access functions, 3-9
    memory for allocating data structures, 3-7
    MITE DMA, 3-7
    NI-VXI software compatibility, 3-9
    NI-VXI software enhancements, 3-9
    number of MITE windows, 3-7
    programming with VXI, 3-3
        example programs (table), 3-4
    shared memory, 3-8
    system configuration functions, 3-9
    window mapping, 3-6

B

BINARY_COMPATIBLE compiler symbol, 3-5

C

capability codes
    VMEbus, A-2
CLK10
    routing (figure), C-5
    routing, VXI bus, C-4
    signal type, E-4
common questions, E-1
    Configuration EEPROM is Invalid message, E-3
    LEDs, definitions of, E-3
    NI-VXI utility programs, E-2
    Resman, E-1
        using, E-2
    serial number and hardware revision of VXI-8340 series, E-1
    shared memory and dual-ported memory, E-4
    VME devices, E-2
    VXI-8340 series system expansion, E-1
compiler symbols for NI-VXI, 3-5
ComponentWorks, 1-5
ComponentWorks++, 1-5
configuration, 2-1, 3-1
    advanced hardware settings, C-1
    advanced options, 1-3, C-1
    default settings, B-1
    EEPROM, C-4
    hardware, 2-6
    setup, 2-1
    software, 2-6
    VXI-8340 series default configuration settings (figure), C-2
    VXIbus slot configuration (figure), C-3
Configuration EEPROM is Invalid message, E-3
conventions used in the manual, *x*
customer education, E-1

D

debugging applications, 3-10
default settings, B-1
  hardware, 2-6, B-1
  software, B-1
device interaction, 3-2
device tab default settings (table), B-1

E

EEPROM
  configuration, C-4
  Configuration EEPROM is Invalid message, E-3
  operation (figure), C-4, C-5
EEPROM operation (figure), C-4
electrostatic discharge (caution), 2-3

F

FAIL LED, 1-3, E-3
front panel
  LEDs, definitions of, E-3
front panel features, 1-3

G

getting started
  what you need, 1-1
go online, 2-7

H

hardware
  advanced configuration settings, C-1
  default settings, B-1, C-1
  description, 1-2
  installing, 2-3
  PCI-8330 series, 1-2
high-level VXIbus access functions, 3-9
how to use manual set, *ix*

I

installation, 2-1
installation and configuration
  configuration utilities, 3-1

L

LabVIEW, 1-5
LabWindows/CVI, 1-5
LEDs
  ACC, 1-3, E-4
  definitions of, E-3
  FAIL, 1-3, E-3
  LINK 1, 1-3, E-3
  LINK 2, 1-3, E-3
  ONLINE, 1-3, E-3
  SYSF, 1-3, E-3
  TX/RX, 1-3, E-4
LINK 1 LED, 1-3, E-3
LINK 2 LED, 1-3, E-3
local resource access functions, 3-10
low-level VXIbus access functions, 3-9

M

MAX, 2-6
  configuration view, 3-3
  device tab default settings (table), B-1
  online help, 3-2
  overview, 1-4, 3-1
  PCI tab default settings (table), B-3
  running Resman, 3-2
  settings for low-level functions and memory sharing (note), 3-4
  shared memory tab default settings (table), B-2
  SMB tab default settings (table), B-3
  viewing device-specific information, 3-2
  VXI bus tab default settings (table), B-2
Measurement and Automation Explorer. *See* MAX

MITE DMA, 3-7
MITE windows, 3-7
MXI IN port, 1-3
MXI OUT port, 1-3

N

National Instruments application software, 1-5
National Instruments Web support, E-1
NI Developer Zone, E-1
NI Spy utility, 1-4, 3-10
NI-VISA/NI-VXI software
    bus interface software, 1-1
    configurations, 1-4
    example programs (table), 3-4
    installing, 2-1
    NI-VXI compiler symbols, 3-5
    overview, 1-4
    programming with, 3-3
    software enhancements, 3-9

O

ONLINE LED, 1-3, E-3
overview
    VXI-8340 series kit, 1-2

P

PCI tab default settings (table), B-3
PCI-8330 series, 1-2
    interface board
        installing, 2-3
    specifications, A-1
PCI-based MXI-3 interface
    software description, 1-4
        National Instruments application software, 1-5
        software configurations, 1-4
power sequencing requirements, D-1
programming. *See* application development

R

related documentation, *xi*
request offline, 2-7
Resman, 2-6
    reconfiguring after power cycling, 3-1
    running through MAX, 3-2
    using, E-2
resource manager. *See* Resman
restart system power, 2-5

S

setup, 2-1
shared memory, 3-8
shared memory tab default settings (table), B-2
slot 0/non-slot 0, C-2
slot configuration, VXI bus (figure), C-3
SMB tab default settings (table), B-3
software
    installation, 2-1
    installation (note), (caution), 2-2
software configuration and verification, 2-6
software configurations, 1-4
software description
    National Instruments application software, 1-5
    NI-VISA/NI-VXI bus interface software, 1-4
specifications, A-1
    environmental, A-1
    physical, A-2
    power requirement, A-1
    requirements, A-1
SYSF LED, 1-3, E-3
system configuration functions, 3-9
system controller, 2-4
system integration, by National Instruments, E-1

T

technical support resources, E-1
TX/RX LED, 1-3, E-4

U

user configurable settings, C-1

V

verification
    software, 2-6
VIC utility, 3-2
VISAIC utility, 3-2
VMEbus capability codes, A-2
VXI bus tab default settings (table), B-2
VXI programming, 3-3
VXI-8340 series
    default configuration settings (figure), C-2
    front panel features, 1-3
    hardware default settings (table), B-1
    interface kit overview, 1-2
    interface module, installing, 2-4
    module description, 1-2
    specifications, A-1
    system expansion, E-1
    typical system setup (figure), 2-3
VXIbus
    CLK10 routing, C-4
    slot 0/non-slot 0, C-2
    slot configuration (figure), C-3
vxiin command, 3-3
vxiout command, 3-3

W

Web support from National Instruments, E-1
window mapping, 3-6
worldwide technical support, E-2